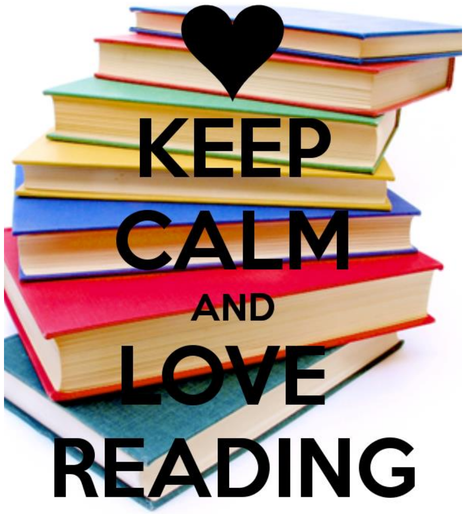
DNS Security

Information Security
CS 526

Omar Chowdhury

Reading for This Lecture

- Optional:
  - First attack by Schuba and Spafford - http://www.openbsd.org/advisories/sni_12_resolverid.txt
  - **An Illustrated Guide to the Kaminsky DNS Vulnerability**
  - Dan Kaminsky's Black Hat presentation (PowerPoint)



Purpose of Naming

- Addresses are used to locate objects
- Names are easier to remember than numbers
- You would like to get to the address or other objects using a name
- **DNS provides a mapping from names to resources of several types**

Domain Name System (DNS)

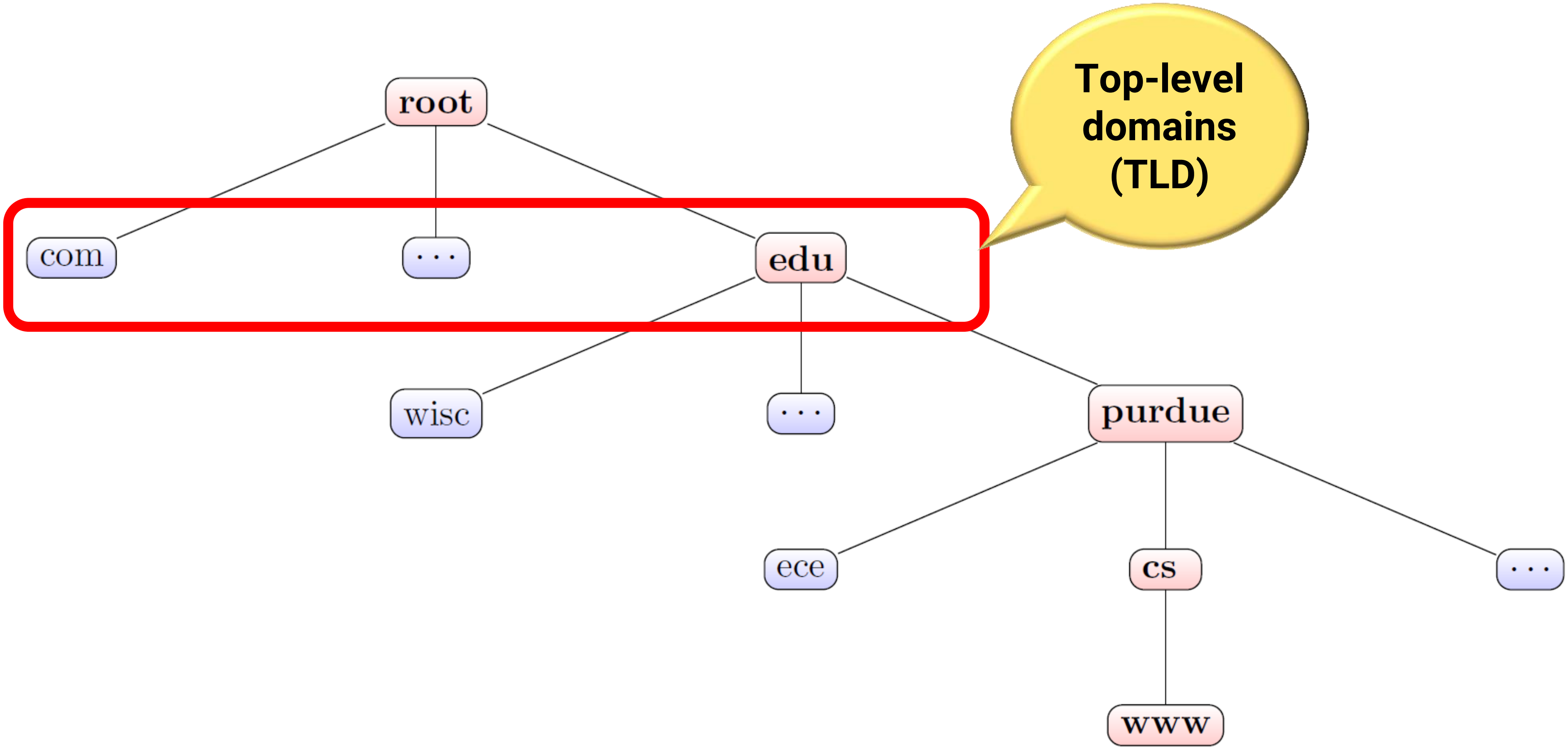
- Forward DNS Resolution
  - Given a domain name lookup the associated IP address
  - **Example**: **cs.purdue.edu ⇒ 128.10.19.20**

- Reverse DNS lookup
  - Given an IP address lookup the domain name associated with the IP address
  - **Usage**: Network troubleshooting, anti-spam techniques

DNS—Distributed Database

- DNS records are kept in a distributed fashion
- Highly dynamic
- Decentralized authority

DNS–Hierarchical Namespace



Root DNS Servers

Domain Name Servers

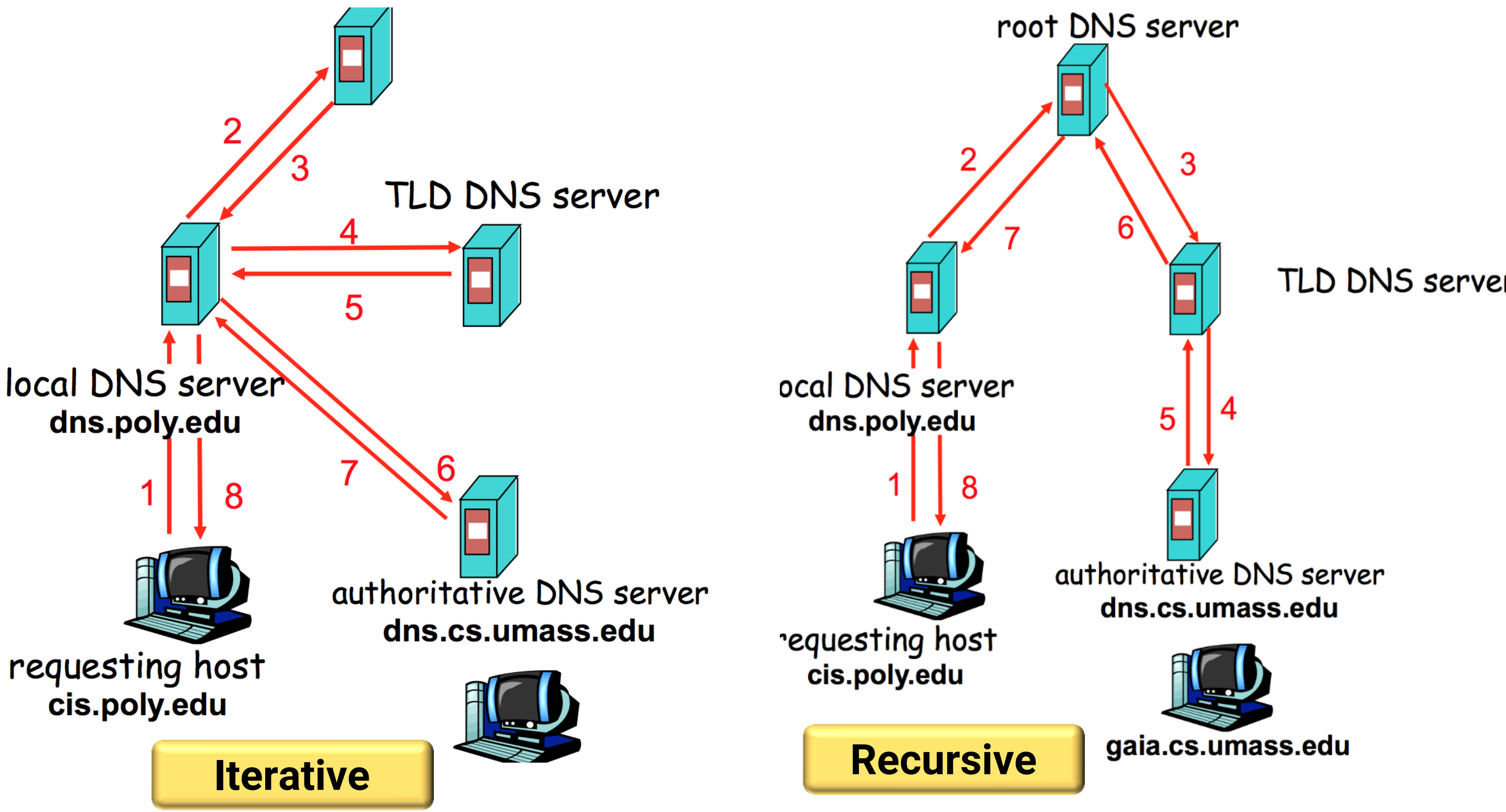
- Top-level domain (TLD) servers
  - Responsible for com, org, net, edu, etc.
  - All top-level country domains, e.g. uk, fr, ca, jp, in.
  - Network Solutions, LLC controls **com** servers
- **cs.purdue.edu's authoritative DNS servers are:**
  pendragon.cs.purdue.edu.
  ns.purdue.edu.
  harbor.ecn.purdue.edu.
  ns2.purdue.edu.
- Local DNS servers
  - Not strictly part of the domain hierarchy
  - Each ISP (university, hospital, company) has one

DNS Resolving

- When a host makes a DNS query, it is forwarded to a local upstream resolver
  - The local upstream resolver acts as a proxy and forwards query into the domain name hierarchy

- Two resolution schemes:
  - Iterative
  - Recursive

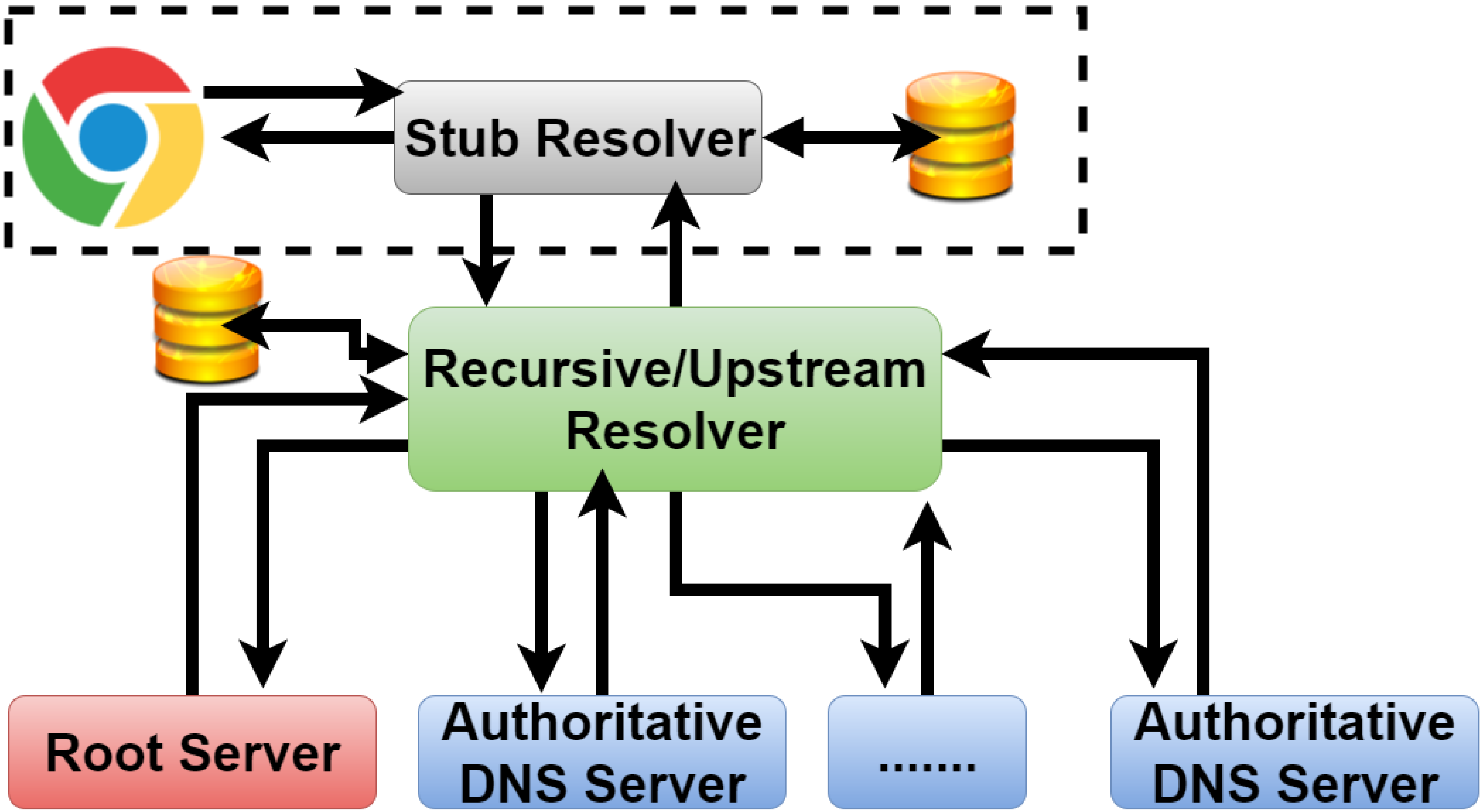
Iterative and Recursive Resolution



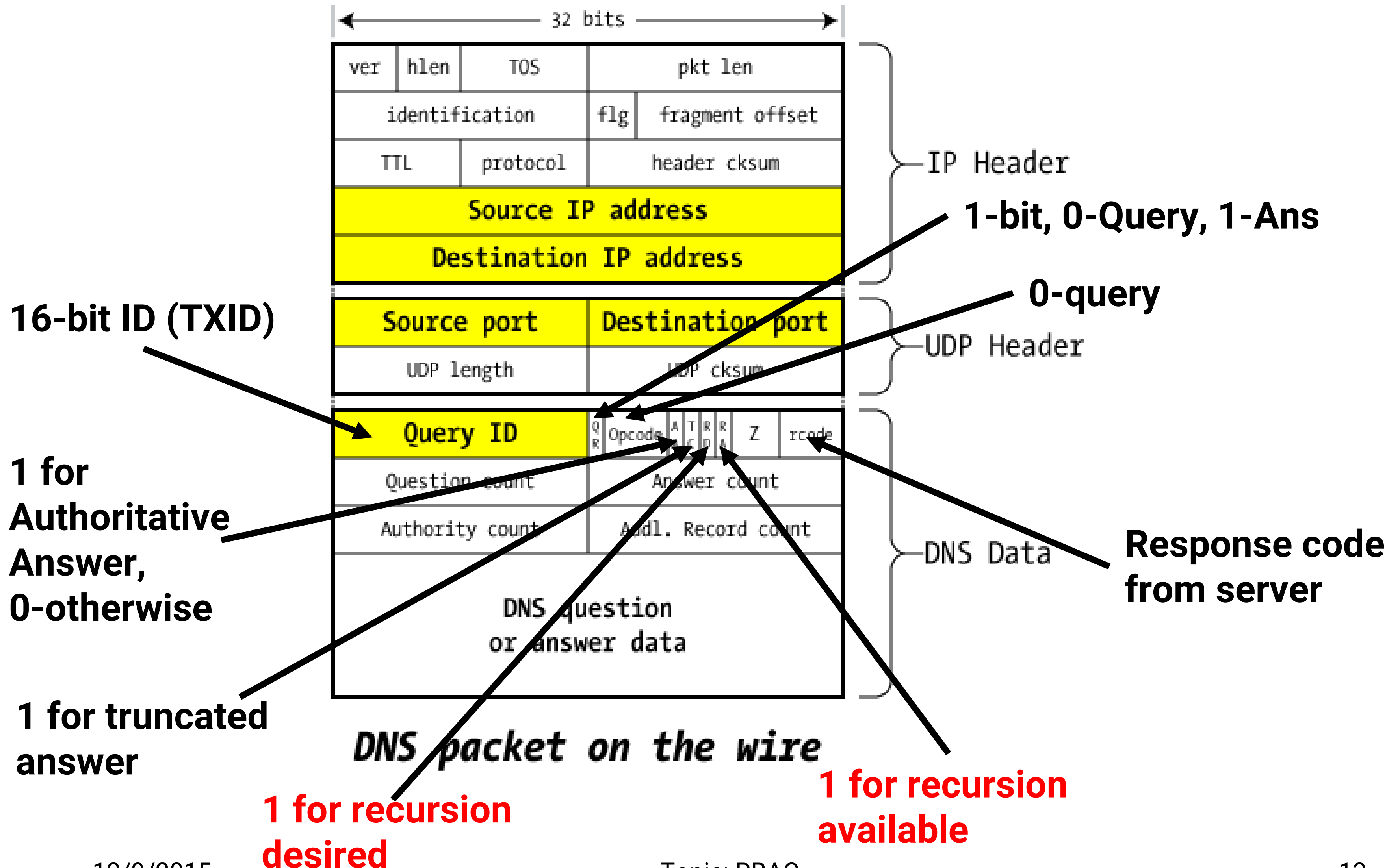
DNS Result Caching

- DNS responses are cached
  - Enables responding to the same query fast
- Negative results are cached too
  - Saves time for nonexistent sites, e.g., mistyping
- Cached data has expiration
  - Each DNS record (also known as, resource record or just RR) has an associated field called Time-To-Live (in short, TTL)

DNS Name Resolution



DNS Packet in the Wire



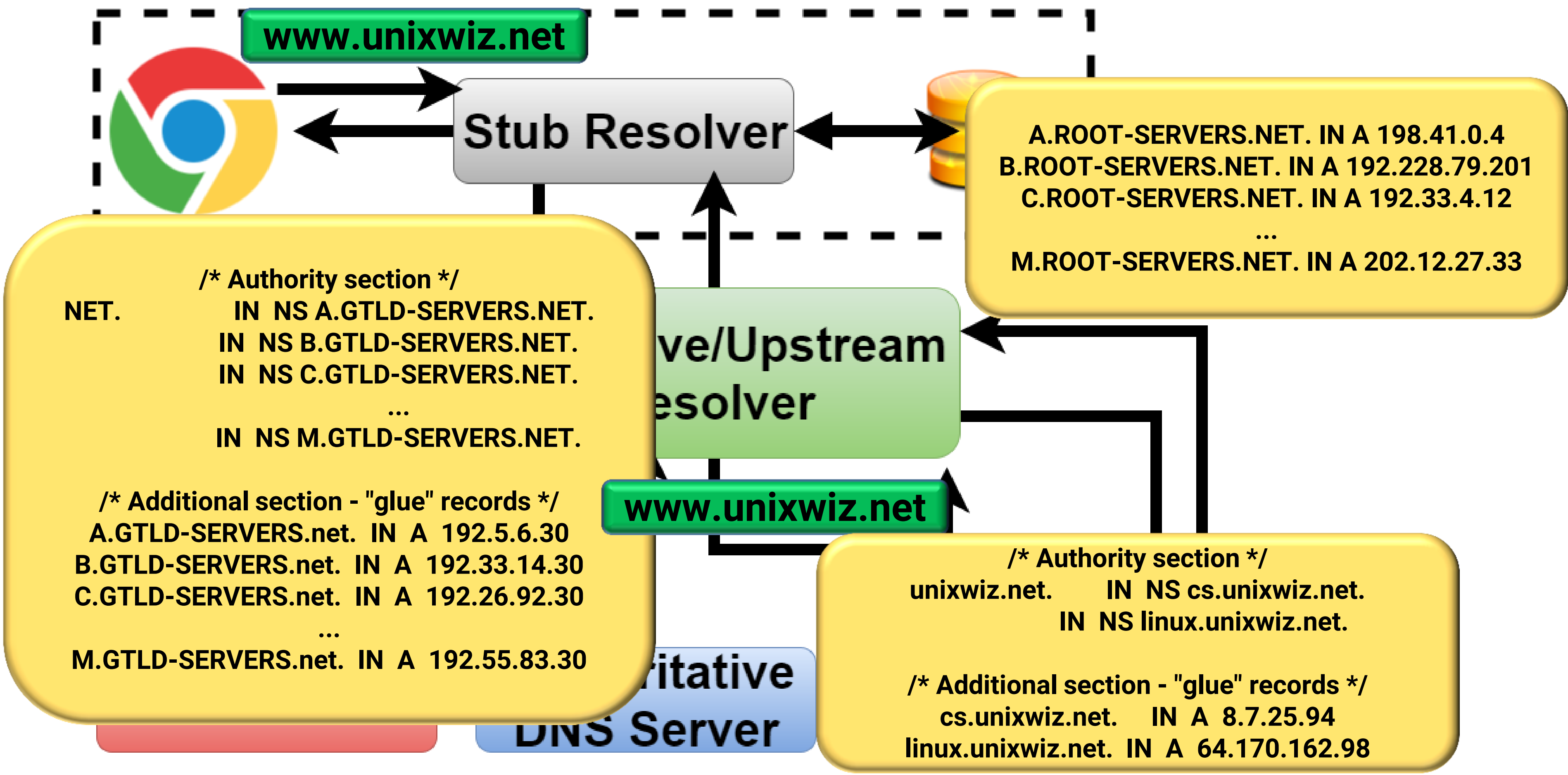
Selective DNS Record Types

- A record
  - Domain name to IP mapping
- NS record
  - Information about (authoritative) DNS server
- MX record
  - Mail exchange record
- SOA record
  - Key information about the zone (e.g., contact address of the admin)
- CNAME record
  - Canonical or alias of a domain
- TXT record
  - Textual description of the domain

DNS Packet Payload

- Question Section
  - Contains the question asked
- Answer Section
  - Contains the records that answer the question
- Authority Section
  - Contains the records that point to domain authority
- Additional Section
  - Glue records
  - IP address of the domain authorities
  - Break circular dependency

DNS Name Resolution



Inherent DNS Vulnerabilities

- Users typically trust the host-address mapping provided by the DNS
  - **What can go wrong with bad DNS information?**
- DNS resolvers trust responses received after sending out queries
  - **How can one exploit this?**
- **Root of all evil**:
  - No authentication for DNS responses

User side attack—Pharming

- Exploit DNS poisoning attack
  - Change IP addresses to redirect URLs to bad sites
  - Potentially more dangerous than phishing attacks

- DNS poisoning attacks are not uncommon:
  - January 2005, the domain name for a large New York ISP, Panix, was hijacked to a site in Australia
  - November 2004, Google and Amazon users were sent to Med Network Inc., an online pharmacy

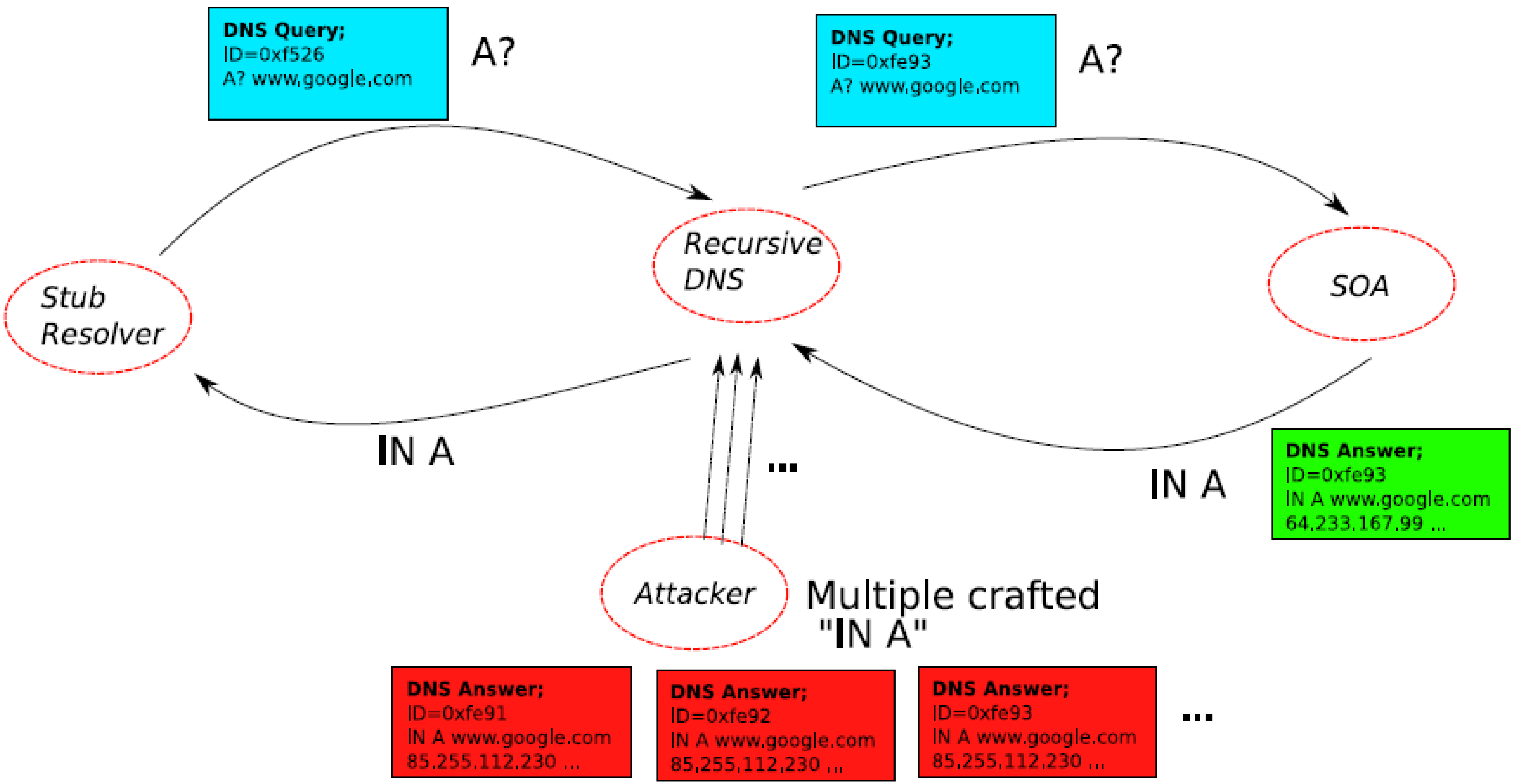
DNS Cache Poisoning

- Attacker wants his IP address returned for a DNS query
- When the resolver asks **ns1.google.com** for **www.google.com**, the attacker could reply first, with his own IP
- **What is supposed to prevent this?**
- Transaction or Query ID
    - 16-bit random number
    - The real server knows the number, because it was contained in the query
    - The attacker has to guess

DNS Cache Poisoning (contd.)

- Responding before the real DNS server
  - An attacker can guess when a DNS cache entry times out and a query has been sent, and provide a fake response.
  - The fake response will be accepted only when its 16-bit transaction ID matches the query
  - CERT reported in 1997 that BIND uses sequential transaction ID and is easily predicted
    - fixed by using random transaction IDs

DNS Cache Poisoning: Racing to Respond First



DNS Cache Poisoning Attack 1

Schuba and Spafford in 1993

- Intuition: Predictable Query ID (QID)
- First, guess QID:
  - Ask (**dns.target.com**) for www.evil.org
  - Request is sent to **dns.evil.org** (get QID)
  - Attacker controls **dns.evil.org**

- Second, attack:
  - Ask (**dns.target.com**) for www.yahoo.com
  - Gives responses from **dns.yahoo.com** to the attackers chosen IP

Defense: the Bailiwicks Rules

- The bailiwick system prevents **foo.com** from declaring anything about **com**, or some other new TLD, or www.google.com

- Using the bailiwicks rules
  - The root servers can return any record
  - The com servers can return any record for **com**
  - The **google.com** servers can return any record for **google.com**

DNS Cache Poisoning Attack 2
Birthday Attack – Vagner Sacramento 2002

- Have many clients send the same DNS request

Defense– Rate limiting
For all the client queries asking for the same domain name, only send out one DNS query

DNS Cache Poisoning – so far

- Early versions of DNS servers deterministically incremented the ID field
- Vulnerabilities were discovered in the random ID generation
  - Weak random number generator
  - The attacker is able to predict the ID if knowing several IDs in previous transactions
- Birthday attack
  - 16- bit (only 65,536 options).
  - Force the resolver to send many identical queries, with different IDs, at the same time
  - Increase the probability of making a correct guess

DNS Cache Poisoning Attack 3
Dan Kaminsky 2008

- Kaminsky Attack
  - Big security news in summer of 2008
  - DNS servers were quickly patched to defend against this attack
  - **Sophisticated attack**
- In prior attacks, when the attacker loses the race, the record is cached with a TTL
  - Before TTL expires, new instance of the attack cannot be carried out
  - Poisoning address for **google.com** in a DNS server is not easy

Features of Kaminsky Attack

- The attacker does not need to wait to launch an attack
- The bad guy asks the resolver to look up **www.google.com**
  - If the bad guy lost the race, the other race for **www.google.com** will be suppressed by the TTL
- If the bad guy asks the resolver to look up **1.google.com**, **2.google.com**, **3.google.com**, and so on
  - Each new query starts a new race
- Eventually, the bad guy will win
  - he is able to spoof **183.google.com**
  - So what? No one wants to visit **183.google.com**

Kaminsky-style Poisoning

- A bad guy who wins the race for "**183.google.com**" can end up stealing "**www.google.com**" as well
- Original malicious response:
  - **google.com NS www.google.com**
  - **www.google.com A 6.6.6.6**
- Killer response:
  - **google.com NS ns1.google.com**
  - **ns1.google.com A 12.34.56.78 [GLUE RECORD]**

Kaminsky-style Poisoning

- Why does it succeed?
  - The attacker can start a new instance of attack anytime without waiting for the cache entry to expire
  - No wait penalty for racing failure
  - The attack is only bandwidth limited

Defenses based on increasing the entropy

Defense – 1 (Source Port Randomization)

- Use a random source port for each out-going DNS query
- **Entropy**: 16-bit (from TXID) + 11-bit (from source port) = 27-bit entropy
- To win, the attacker has a much bigger space of possible IDs to forge a valid response
- **Limitation**: When the resolver is behind a NAT device, the additional entropy provided by source port randomization is low (e.g., 2 bits)

Defense – 2 (0x20 Randomization)

- Domain names are case-insensitive
- www.google.com and WWW.GOOGLE.COM resolves to the same IP address
- Randomly change some of the characters in the domain name to upper case letters
- The expectation is that the resolver responding will copy the domain name in the response from the query
- **Limitations**:
  - Attacker can query: www.123.com
  - Some of the resolvers always return domain name in all lower-case
  - Some of the resolvers use a case-sensitive matching
  - 70% resolvers support 0x20 randomization

Defense – 3 (WSEC DNS)

- While querying root or TLD DNS servers, preprend random prefix
- **Ddsj030gojfd.www.google.com** and www.google.com returns the same RRs (not IP)
- **Limitations**:
  - Domain names can be maximum 255-byte length
  - Attackers can query 255-byte domains
  - This is applicable to referrals not A queries
  - Some resolvers exclude long domain names as they are uncommon

Defense – 4
(Randomize destination IP address)

- Multiple IP addresses for a DNS server
- Choose one randomly from that list to query

ns.purdue.edu.	86400	IN	A	128.210.11.5
ns2.purdue.edu.	86400	IN	A	128.210.11.57
harbor.ecn.purdue.edu.	86400	IN	A	128.46.154.76
pendragon.cs.purdue.edu.	86400	IN	A	128.10.2.5

  - They are many time predictable
  - Attack it to make it more predictable

Defense—5 (Randomizing Source IP)

- This is called the NAT-antidote
- This is applicable when the resolver is behind a NAT
- When NAT receives a query from the resolver, it randomly changes the source IP address to a random IP in that network
- **Limitations**:
  - Extra entropy depends on the network size

Adaptive & Long-term Defenses

Attack Detection

Received a response for a query and the TXID of the response does not match with the query's TXID

Can happen in benign cases as DNS runs on top of UDP

Defense—6 (Sandwich Antidote)

- Detect attack and apply sandwich antidote
- For querying www.google.com
  - Query 1: dfjfdkjfhksdf.google.com
  - Query 2: www.google.com [REAL QUERY]
  - Query 3: jkjkjoiojohh.google.com
- Observe whether the response arrive in-order
- **Limitation**:
  - Our experiment suggest that 50% of the queries arrive in order

Defense—7 (DNSSEC)

- Proposed as a long-term solution
- Uses digitally signed responses
- Prevents cache poisoning attacks
- Limitations:
  - Adoption is very slow– 1% of all the .com and .net domains are secured by DNSSEC
  - Opens door for DoS attack due to large response size

Defense – 8 (Use TCP)

- Run DNS protocol on top of TCP instead of UDP
- **Limitations**:
  - High latency (almost 2X)
  - Resolver throughput rate ($1/10^{th}$)
  - Not all resolvers support TCP
  - In our experiment with Alexa's top 15K domains, 10% of their authoritative nameservers do not support TCP – includes **Facebook**

Defense – 9 (Adaptively use TCP)

- Run DNS protocol on top of TCP instead of UDP **only when detected attack**
- Nominum implemented it in their product Vantio CacheServe
- **Limitations**:
  - Attack detection is not fine-grained – benign cases will be considered as attack
  - Same as before
  - Susceptible to the new attack we have proposed

Acknowledgement

Some of the slide materials are inspired by materials from Ninghui Li and http://unixwiz.net/.